---

### Diabetes Risk Detection

**Title:**
**Health Risk Predictor: Diabetes Risk Detection using Machine Learning (SDG 3 – Good Health & Well-being)**

**Student Name: Samuel Kamawira**
**Course / Unit: AI For Software Engineering**

---

# 1. Introduction

Non-communicable diseases such as diabetes are a major public health challenge worldwide. Many people do not know they are at risk until the disease has already progressed, which can lead to complications like heart disease, kidney failure, and vision problems. Early awareness and screening are therefore essential.

This project implements a simple **Health Risk Predictor** that estimates the risk of type 2 diabetes using basic health parameters (such as age, BMI, blood pressure, and blood glucose). The system uses a machine learning model trained on a public dataset and provides a user-friendly interface built with Streamlit.

The project directly supports **Sustainable Development Goal 3 (SDG 3): Good Health and Well-being** by:

- Raising **awareness** of diabetes risk.
- Encouraging **early screening** and preventive action.
- Demonstrating how low-cost, offline digital tools can support health decision-making.

> **Important note:** This system is for educational purposes only and is **not** a substitute for professional medical diagnosis.

---

# 2. SDG 3 Relevance

**SDG 3: Good Health and Well-being** aims to “ensure healthy lives and promote well-being for all at all ages.” One of its targets (SDG 3.4) focuses on reducing premature mortality from non-communicable diseases (NCDs) such as cardiovascular disease, cancer, diabetes, and chronic respiratory disease.

This project contributes to SDG 3 in the following ways:

- **Awareness of non-communicable diseases:** The app helps users understand that factors like glucose level, BMI, age, and blood pressure influence their diabetes risk.

- **Early screening mindset:** Even though the model is simple, showing a “risk estimate” can motivate users to seek proper medical screening and adopt healthier lifestyles.

- **Accessibility and affordability:** The entire system runs **offline** on a normal laptop, without paid APIs or internet connectivity. It can be used in low-resource or low-connectivity environments such as rural clinics or community health campaigns.

By combining data, AI, and a simple user interface, the project shows how digital tools can support preventive healthcare and health education.

---

# 3. Dataset Description

The model is trained on the **Pima Indians Diabetes Dataset**, a well-known benchmark dataset used in many academic examples of medical machine learning.

**Key characteristics:**

- **Number of records:** 768

- **Target variable:**

    1. `Outcome` (0 = no diabetes, 1 = diabetes)

- **Input features (8):**

    1. **Pregnancies** – Number of times pregnant

    2. **Glucose** – Plasma glucose concentration (mg/dL)

    3. **BloodPressure** – Diastolic blood pressure (mm Hg)

    4. **SkinThickness** – Triceps skinfold thickness (mm)

    5. **Insulin** – 2-Hour serum insulin (mu U/ml)

6. **BMI** – Body Mass Index (weight/height²)
7. **DiabetesPedigreeFunction** – A score related to family history of diabetes
8. **Age** – Age in years

### 3.1 Data Preprocessing

Some physiological features in the dataset contain **0 values**, which are not realistic in a medical context (for example, blood pressure of 0). In this project:

- The following columns were treated as having missing values when equal to 0: `Glucose`, `BloodPressure`, `SkinThickness`, `Insulin`, `BMI`.
- These zeros were replaced with **NaN** and then imputed using the **median value** of each column.
- This simple imputation approach keeps the dataset size the same while handling unrealistic measurements.

After preprocessing, the dataset is ready to be used for model training and evaluation.

---

## 4. Methodology

### 4.1 Tools and Libraries

The project was implemented in **Python**, using the following libraries:

- **pandas, numpy** – Data loading and preprocessing
- **scikit-learn** – Machine learning (model training, evaluation, scaling)
- **pickle** – Saving the trained model and scaler
- **Streamlit** – Building the web-based user interface

### 4.2 Train–Test Split

To evaluate the model, the dataset was split into:

- **80% training data**

- **20% test data**

A **stratified split** was used to preserve the proportion of diabetic vs non-diabetic cases in both training and test sets.

### 4.3 Feature Scaling

Because the features have different units and ranges, the input data was scaled using **StandardScaler**:

- Each feature is transformed to have **zero mean** and **unit variance**.
- Scaling improves the performance of models like Logistic Regression and ensures that no single feature dominates purely due to its scale.

The scaler is fitted on the training data and then applied to both training and test sets, and later to new user input in the app.

### 4.4 Model Selection

The chosen model is **Logistic Regression**:

- It outputs the **probability** of the positive class (diabetes) between 0 and 1.
- It is interpretable and computationally efficient.
- It is suitable for binary classification tasks and is commonly used in medical risk models.

Other models (such as Decision Trees or Random Forests) could be used, but Logistic Regression was selected because it:

- Keeps the project simple and lightweight.
- Produces a small model file that is easy to deploy.
- Provides meaningful probabilities that can be turned into risk levels (Low/Medium/High).

### 4.5 Model Saving

After training, the following objects are saved to a file called **`diabetes_model.pkl`** using `pickle`:

- The trained Logistic Regression **model**
- The **StandardScaler** used for feature scaling
- The list of **feature names**

This artifact is later loaded by the Streamlit app to make predictions offline.

---

# 5. Results and Evaluation

> *(Fill in this section with your own numbers from running `train_model.py`.)*

After training, the model was evaluated on the test set (20% of the data). The following are example metrics (replace with your actual results):

- **Test Accuracy:** ~0.78 (78%)
- **Precision (diabetes class):** ~0.74
- **Recall (diabetes class):** ~0.68

These results show that the model can reasonably distinguish between people with and without diabetes in this dataset, although it is not perfect.

**Interpretation:**

- **Accuracy** indicates that about 78% of test samples were correctly classified.
- **Recall for the diabetes class** is important because false negatives (people who have diabetes but are predicted as low risk) are more dangerous in a health context.
- In real healthcare applications, models should be optimized to minimize false negatives and need extensive validation.

The results are acceptable for a **student project and awareness tool**, but not enough for clinical use.

---

# 6. Application Design & Deployment

## 6.1 Streamlit User Interface

The trained model is integrated into a **Streamlit web application** called `app.py`. The app has three main tabs:

1. 🔮 **Risk Prediction**
   - Sidebar allows the user to input:
     - Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age.
   - The app:
     - Scales the input using the saved scaler.
     - Uses the Logistic Regression model to predict diabetes risk.
     - Displays:
       - A binary prediction (low risk vs diabetes risk).
       - The estimated probability of diabetes (0–1).
       - A **risk level** (Low / Medium / High) based on probability.
     - Shows general, non-medical **health tips** promoting diet, physical activity, and check-ups.
     - Includes clear disclaimers that this is not a medical diagnosis.
2. ℹ️ **About & SDG 3**
   - Explains:
     - The goal of the project.
     - The connection to **SDG 3: Good Health & Well-being**.
     - A short description of the dataset and model.
     - An explicit **disclaimer** about non-clinical use.
3. 📈 **Usage Stats**
   - Reads a local CSV file (`usage_log.csv`) which stores anonymous logs of past predictions.
   - Shows:

        - Total number of predictions made.
        - Average predicted probability of diabetes.
        - A bar chart of risk level distribution (Low/Medium/High).
        - A table of the last 10 predictions.

### 6.2 Offline Deployment

The system is designed to work **completely offline** after installation:

- All code (`train_model.py`, `app.py`) and data (`diabetes_model.pkl`, `diabetes.csv`, `usage_log.csv`) are stored locally.
- No external API calls are required.
- Users simply run:

```
streamlit run app.py
```

- The app opens in a web browser at `http://localhost:8501`.

This makes the app suitable for:

- Classroom demonstrations
- Offline health awareness sessions
- Environments with limited or no internet connectivity

The project folder can be zipped and shared, and any user with Python and the required libraries installed can run the app.

---

## 7. Logging and Simple Analytics

To make the project more realistic, each prediction is logged (without names or personal identifiers) to **`usage_log.csv`**:

- For every prediction, the app saves:

    - Timestamp
    - Input features
    - Predicted class (0 or 1)
    - Predicted probability
    - Risk level (Low/Medium/High)

The **Usage Stats** tab uses this file to:

- Show summary statistics.
- Visualize how many predictions fall into each risk level.
- Display the most recent predictions.

This illustrates how simple analytics can be integrated into AI health tools, which can be useful in monitoring tool usage and understanding typical risk patterns in outreach scenarios.

---

# 8. Limitations

Despite its usefulness as a learning project, the system has several important limitations:

1. **Dataset limitations:**
    - The Pima Indians Diabetes dataset focuses on a specific population (Pima Indian women), which may limit how well the model generalizes to other groups.
2. **Limited features:**
    - Only eight numerical features are used. Real clinical diagnosis would involve more detailed history, lab tests, and physical examinations.
3. **Model simplicity:**
    - Logistic Regression is simple and interpretable but not necessarily the most accurate model for this problem. More advanced models (e.g. Random Forests, XGBoost) could perform better.
4. **No clinical validation:**

- The model has not been tested or validated in a real clinical environment.
- It is not intended for actual medical decision-making.

Because of these limitations, the app must be viewed as an **educational and awareness tool**, not a diagnostic system.

---

## 9. Future Work

Potential improvements include:

1. **Additional Models and Comparison:**
   - Train and compare more advanced models (Random Forest, Gradient Boosting, etc.) and choose the best based on metrics such as recall, precision, and AUC.
2. **Better Interpretability:**
   - Use techniques like feature importance plots, SHAP values, or explanations that show how each feature contributed to a specific prediction.
3. **Multi-disease Risk Prediction:**
   - Extend the app to include other conditions (e.g. heart disease risk) on separate tabs, turning it into a small "NCD risk dashboard."
4. **Improved User Guidance:**
   - Add more educational content on healthy lifestyle choices and links to trusted health resources.
5. **Localization and Language Support:**
   - Provide the interface in multiple languages to increase accessibility.

---

## 10. Conclusion

This project successfully demonstrates how **machine learning** and a simple **web interface** can be combined to create an offline **Health Risk Predictor** related to type 2 diabetes. Using the Pima Indians Diabetes dataset and a Logistic Regression model, the system estimates diabetes risk from basic health parameters and visualizes the results in a user-friendly way.

By promoting awareness of diabetes and encouraging early screening, the project aligns with **SDG 3: Good Health & Well-being**, specifically the reduction of premature mortality from non-communicable diseases. Although the model is not a clinical tool, it serves as a valuable educational example of how AI can support public health and preventive care.